Amendments to the Claims

The following listing of claims will replace all prior versions and listing of claims in the application.

Claims 1-26 (Canceled)

26. (currently amended) A method for debugging a ~~programs~~ program for an industrial ~~controllers~~ controller, the industrial controller having an engineering system including an editor and further having a run time system, wherein graphical elements are linked using ~~an~~ the editor to form a motion control flowchart that can be visualized on a display, the graphical elements corresponding to respective tasks, the method comprising the steps of:

a) preparing a debugging process based on the flowchart;

b) assigning a suspend command to each ~~graphic~~ graphical element;

c) commencing the debugging process;

d) continuing the debugging process until a suspend command is reached;

e) displaying the location of the flowchart element corresponding to the ~~suspended~~ suspend command; and

f) proceeding to the next possible suspend command[.];

wherein a task corresponding to a graphical element of the flowchart, that has been suspended by a suspend command, is continued by a task control mechanism of the run-time system.

Claim 27 (canceled)

Claim 28 (canceled)

29. (currently amended) The method according to claim ~~27~~ 26, wherein the task control mechanism of the run time system comprises breakpoint debugging and ~~variable~~ variables that can be pre-assigned by the user in the engineering system, further comprising the step of pre-assigning variables corresponding to breakpoints.

30. (previously presented) The method according to claim 29 wherein the variable pre-assignments in the task control mechanism are performed by programs of the run time system other than the task control mechanism.

31. (currently amended) The method according claim 26, comprising the steps of:
a) generating a structured textual language from the flowchart;
b) converting the structured textual language into a processor-independent pseudo-code;
c) loading the processor-independent pseudo-code into ~~the~~ a controller;
d) converting the processor-independent pseudo-code into executable processor code.

32. (previously presented) The method according to claim 26, wherein a debugging interface is available to a user at ~~the level~~ levels comprising ~~one~~ at least two of the group consisting of the structured textual language level, the pseudo-code level, and the processor code level.

33. (currently amended) A method according to claim 26, wherein programming language commands are provided in the flowchart editor as a function of configuration of hardware associated with ~~the~~ an industrial controller.

34. (currently amended) The method according to claim 26, wherein additional graphical elements are generated in the motion control flowchart representation by converting user-defined structured text subprograms of the textual language ~~the~~ into graphical elements comprising function interfaces of the corresponding structured text subprograms.

35. (previously presented) The method according to claim 34, wherein the generated graphical elements are used as language elements of the motion control flowchart.

36. (currently amended) The method according to claim 26, wherein structured text according to IEC 6-1131 is used as ~~the~~ a structured textual language.

37. (previously presented) The method according to claim 36, wherein a user can switch between structured textual language, contact plan and function plan as forms of representation for formulation conditions.

38. (previously presented) The method according to claim 26, wherein at least one of the group consisting of a loop and a parallel branch is present as a programming language command in the motion control flowchart view.

39. (currently amended) The method according to claim 38, wherein a parallel branch is initiated ~~an~~ and wherein individual commands are initiated in a given interpolator cycle within a respective parallel branch.

40. (currently amended) The method according claim 26, ~~wherein~~ further comprising function blocks, wherein parameters can be set for the function blocks by mask input in a display associated with the motion control flowchart ~~view~~.

41. (currently amended) The method according to claim 26, ~~wherein~~ further comprising function blocks, wherein the function blocks are combined into modules that in turn are presented as function blocks in a display associated with the motion control flowchart ~~view~~.

42. (currently amended) The method according to claim 41, wherein modules are interleaved in the display associated with the motion control flowchart ~~view~~.

43. (currently amended) The method according to claim 41, wherein the function blocks for the allocation of variables in the display associated with the motion control flowchart ~~view~~ comprise multiple instructions.

44. (currently amended) The method according to claim 41, wherein the function blocks representing functions that require a given period of time comprise advance conditions in the display associated with the flowchart ~~view~~.

45. (previously presented) The method according to claim 26, wherein the graphical elements of the flowchart are positioned automatically.

46. (previously presented) The method according to claim 26, wherein the graphical elements of the flowchart are linked together automatically.

47. (currently amended) The method according to claim 26, wherein the flowchart is displayed in a form comprising one of the group consisting of a reduced form and ~~on or~~ an enlarged form.

48. (previously presented) The method according to claim 31, wherein re-translation back into motion control flowchart representation is possible by means of marks in the textual language.

49. (previously presented) The method according to claim 26, wherein steps a) through c) are triggered in a collective step.

50. (previously presented) The method according to claim 26, wherein during processing of the flowchart program a currently processed graphical element is displayed.

51. (new) The method according to claim 26, wherein the different code levels comprise at least one of the group consisting of structured text code, processor independent pseudo-code, and object code.

52. (new) A method for debugging a program for an industrial controller, the industrial controller having a plurality of code levels associated with at least one of an engineering system and a run time system associated with the industrial controller, wherein graphical elements are

linked using an editor to form a motion control flowchart that can be visualized on a display, the method comprising the steps of:

a) preparing a plurality of debugging processes for programming code associated with the flowchart;

b) conducting debugging for the plurality of debugging processes; and

c)displaying the debugging processes on respective ones of a plurality of debugging interfaces.

53. (new) The method according to claim 52, wherein the programming code comprises a plurality of code levels, at least a subset of the plurality of debugging processes corresponds to respective ones of the plurality of code levels, and the step of displaying debugging processes comprises displaying at least a subset of the debugging processes on respective ones of the plurality of debugging interfaces.

54. (new) The method according to claim 52, wherein the plurality of code levels comprises a pseudo code level and a debugging process is prepared for the pseudo code level.

55. (new) A method for debugging a program for an industrial controller, the industrial controller having a plurality of code levels associated with at least one of an engineering system and a run time system associated with the industrial controller, wherein graphical elements are linked using an editor to form a motion control flowchart that can be visualized on a display, the method comprising the steps of:

a) preparing a debugging process for programming code associated with the flowchart;

b) conducting debugging for the debugging process; and

c) displaying the debugging process at a plurality of levels.

56. (new) The method according to claim 55, wherein the plurality of levels comprise at least one selected from the group consisting of the structured textual language level, the pseudo-code level, and the processor code level.